

ООО «Компания Семь Печатей»

117216, Москва, ул. Феодосийская, д. 1, корп. 6; тел.(факс): (495)225-25-31, (495)020-23-46

Email: 2252531@mail.ru; Web-page: www.sevenseals.ru, www.shop-sevenseals.ru



Драйвер оборудования СКУД

TSS-2000
NTService

SDK

РУКОВОДСТВО РАЗРАБОТЧИКА

Москва

2018

Оглавление

1. Назначение драйвера оборудования.....	2
2. Системные требования.....	2
3. Комплект поставки сервиса.....	2
4. Принципы подключения и функционирования оборудования СКУД	3
4.1. Способы подключения оборудования	3
4.2. Комплексный и автономный режимы работы.....	4
4.3. Краткий обзор событий контроллера и их адресации.....	5
5. Работа с SDK	6
6. Работа драйвера.....	6
6.1. Работа с клиентами.....	6
6.2. Режимы работы с оборудованием.....	7
6.3. Восстановление связи с оборудованием.....	8
7. Система команд и событий.....	8
7.1. Запись протокола «сырых» событий	8
7.2. Запись ключа в память контроллера.....	9
7.3. Работа с расписанием	9
7.4. Работа с реле.....	9
8. Приложение 1 Система команд	11
9. Приложение 2 Система событий	24

В документе используются специальные термины и выражения. Для полного понимания информации, изложенной в данном тексте, рекомендуем ознакомиться с глоссарием «TSS0011_Словарь терминов».

1. Назначение драйвера оборудования

Драйвер оборудования¹ (программный модуль *Servcont*) предназначен для создания программного обеспечения системы контроля и управления доступом, регистрации рабочего времени, комплексных систем безопасности и других систем на базе сетевых контроллеров марки TSS.

Драйвер служит для осуществления двухстороннего обмена данными между контроллерами СКУД марки TSS и компьютером (программным обеспечением). Он предназначен для использоваться в качестве основного компонента (сервера) программного обеспечения, создаваемого сторонними разработчиками.

2. Системные требования

Программа *TSS-2000 Servcont* предназначена для функционирования в среде ОС Windows XP, Windows Vista, Windows 2003, Windows 2008, ОС Linux².

Главные требования к ПК, на котором будет функционировать драйвер – это надежность. Рекомендуется использовать процессор архитектуры Intel. Прочие характеристики особого значения не имеют (кроме случаев с развитой конфигурацией оборудования СКУД, о чем позже) и являются средними для компьютеров, производимых в текущий момент времени.

Естественно, в предыдущем абзаце речь шла о требованиях только к описываемому модулю. Если на этом же ПК будут работать и прикладные программы пользователя, то требования к нему (компьютеру) необходимо корректировать.

При работе с большим количеством оборудования требования к ПК повышаются. Однако разработчиками рекомендуется схема, при которой на каждый драйвер оборудования приходится не более 40 – 50 контроллеров (или 300 – 350 пунктов прохода). Т.е. для СКУД, рассчитанных на сотни и тысячи пунктов прохода рекомендуется строить распределенную систему с несколькими серверами оборудования, разнесенных по рабочим станциям ЛВС.

Связь с драйвером оборудования осуществляется по порту 4000.

3. Комплект поставки сервиса

В стандартный комплект поставки сервиса входят:

- Папка Bin, содержащая:
 - Собственно драйвер оборудования (*Servcont.exe*).
 - Служба Windows (*Servconts.exe*), позволяющая стартовать драйвер как системную программу (естественно, только под Windows).

¹ Далее для обозначения драйвера оборудования будут применяться также термины Сервер оборудования, Сервер, программный модуль, программа. В каждом случае подчеркиваются те или иные особенности описываемого предмета.

² Теоретически драйвер должен работать на всех UNIX-подобных системах. На сегодняшний день он протестирован на следующих клонах Linux: uclinux, Suise, Debian.

Набор необходимых для работы библиотек.

- Папка SDK, содержащая:
Библиотеку ActiveX компонентов (AxBFServCont.dll).
Папку Demo, с программой и исходными текстами демонстрационного примера ServontClient (Delphi 7).
- Папка Doc, содержащая:
Настоящую документацию.
Документацию на монтаж оборудования СКУД.

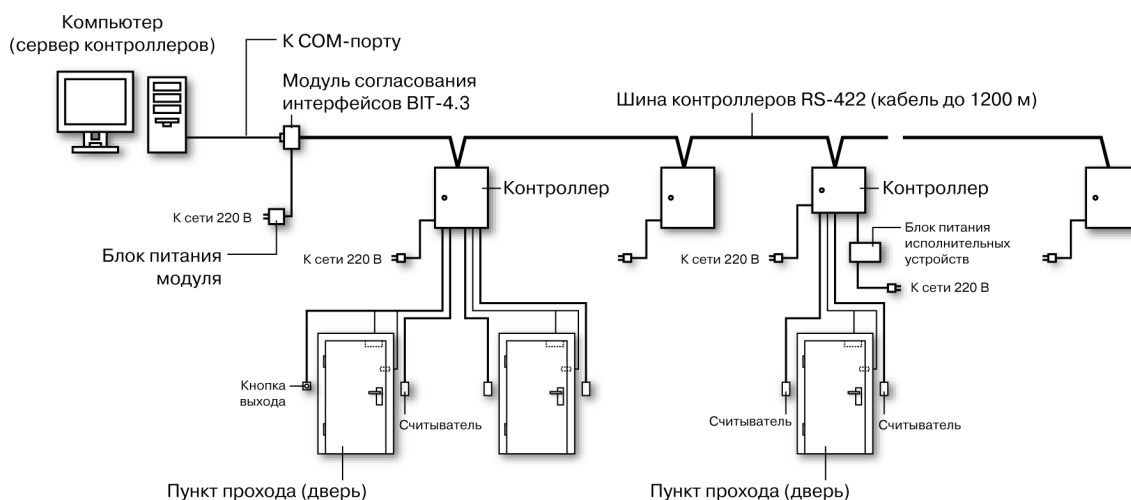
4. Принципы подключения и функционирования оборудования СКУД

Для понимания дальнейшего необходимо дать общее представление об оборудовании СКУД и способах его подключения.

4.1. Способы подключения оборудования

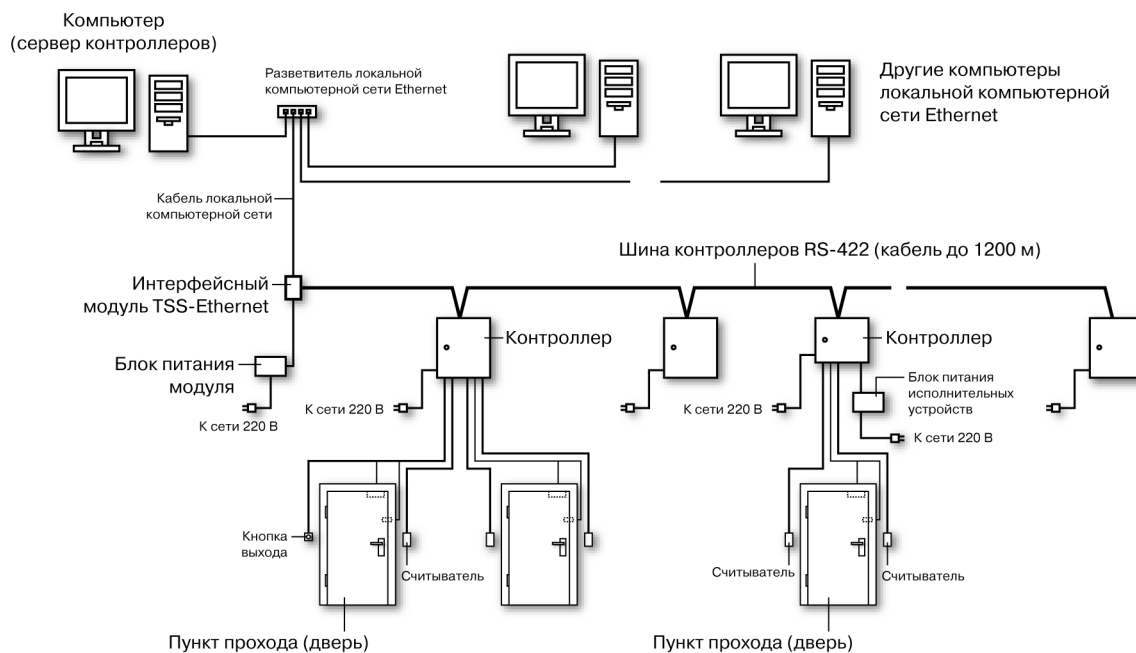
Существует два способа связи сети контроллеров с компьютером, на котором установлен драйвер:

Контроллеры подсоединяются к COM-порту компьютера посредством интерфейсного модуля ВІТ-4.3³.



Контроллеры связываются с компьютером по локальной компьютерной сети, в которой находится компьютер. При этом способе подключения используется интерфейсный модуль типа TSS-Ethernet (RS 232 – Ethernet).

³ Интерфейсный модуль ВІТ-4.3 предназначен для преобразования сигналов интерфейса RS-422 в сигналы интерфейса RS-232.



При задании настроек драйвера необходимо указать способ связи компьютера и контроллеров. Для этого используется понятие т.н. «каналов» (Channels).

Под **каналом** понимается линия связи контроллеров с компьютером, подключенных к компьютеру одним из указанных выше способов (с помощью интерфейсного модуля ВПТ-4.3 или по компьютерной сети). Канал связи указывается заданием либо адреса СОМ-порта (при использовании модуля типа ВПТ-4.3), либо IP-адреса интерфейсного модуля (при использовании модуля типа TSS-Ethernet).

Наиболее надежным способом связи компьютера с контроллерами является их подключение к СОМ-порту. Связь через локальную сеть более подвержена сбоям (как вследствие ограничений сетевой карты интерфейсного модуля, так и по причине проблем, возникающих при работе локальной компьютерной сети, особенно с доменной архитектурой). Этот способ связи рекомендуется использовать только в тех случаях, когда полностью отсутствует возможность подключения контроллеров к компьютеру с помощью интерфейсного модуля ВПТ-4.3 и шины контроллеров (например, при значительном отдалении контроллеров от центра управления).

4.2. Комплексный и автономный режимы работы

Контроллеры СКУД являются достаточно функциональными устройствами и реализуют следующие⁴:

- Хранят до 64025 кодов электронных ключей.
- Хранят до 250000 событий.
- Хранят до 16 расписаний и до 256 праздников.
- Обеспечивают проверку прав доступа приложенного ключа по параметрам:
 - Наличие в базе ключей.
 - Разрешение доступа по данному пункту прохода.
 - Разрешение доступа в данное время.

⁴ Речь идет о контроллерах 209 серии.

Подчеркиваем, что указанные механизмы реализуются самими контроллерами при автономной работе⁵. Тем самым выполняется главное правило работы оборудования любой системы безопасности – возможность выполнения основных функций при потере связи с управляющей программой.

4.3. Краткий обзор событий контроллера и их адресации

Каждый контроллер СКУД обслуживает от 2 до 8 пунктов прохода. Конструктивно пункт прохода – это порт контроллера. Каждый порт имеет три входа от периферийных устройств и один выход на реле. Контроллер принимает данные от следующих устройств:

- Считыватель (ридер) кода электронного ключа.
- Датчик двери (геркон).
- Кнопка открытия двери.

Каждое событие имеет адрес, состоящий из адреса контроллера и номера порта.

Каждое событие снабжено датой и временем своего возникновения (дата и время контроллера).

В событии от считывателя присутствует код ключа.

Датчик двери порождает два события: на размыкание (открытие двери) и на замыкание (закрытие).

Кнопка является нормально разомкнутым устройством, т.е. событие от нее приходит только при замыкании.

Кроме этих событий контроллер генерирует служебные события:

- Отсутствие питания 220 В.
- Падение напряжения на аккумуляторе ниже 12 В.
- Открытие крышки контроллера.

Для управления контроллером существует система команд. Вот основные из них:

- Включить реле порта контроллера.
- Загрузить ключ.
- Загрузить расписание.
- Загрузить чипы (адреса охранных датчиков).

⁵ Естественно, что для этого предварительно должна быть выполнена загрузка ключей и прав доступа посредством внешней программы.

5. Работа с SDK

Перед началом работ по написанию клиентского приложения или проверки поставляемого демо-примера следует зарегистрировать библиотеку *AxBFServCont.dll*.

Для работы в среде программирования достаточно импортировать указанную библиотеку без инсталляции компонентов на палитру. При желании работать с визуальными компонентами имейте в виду, что работа выполняется с компонентом *TSSServContX*.

Рекомендуется выполнять написание программы на основе полнофункционального (т.е. использующего все возможности системы) демонстрационного примера *Servcont_client*.

6. Работа драйвера

6.1. Работа с клиентами

Для прикладной программы драйвер оборудования является сервером. Прикладная программа – клиентом.

Обмен между клиентом и сервером реализован на основе TCP/IP протокола посредством механизма вызова удаленных процедур. Технически он реализован специальным классом разработчика.

Для *Сервера* существует понятие «*Главный клиент*», т.е. клиентское приложение, для которого выполняются два обязательных условия⁶:

- В отсутствие главного клиента все оборудование безусловно переводится в автономный режим.
- Управлять оборудованием может только *Главный клиент*.

Сервер позволяет выполнять соединение с ним произвольному числу клиентов, однако главным из них может быть только один. Остальные имеют право получать информацию о работе системы во время работы *Главного клиента*.

Основные функции драйвера:

- Установление и поддержание устойчивой связи с оборудованием.
- Перевод оборудования в комплексный или автономный режим.
- Передача клиентам событий, возникающих при работе оборудования.
- Прием от клиентов команд, необходимых для выполнения функций СКУД.

Необходимо подчеркнуть, что сам *Сервер* не инициирует **ничего**. После старта он ждет подсоединение *Главного клиента*, который и должен осуществлять управление системой в целом и *Сервером* в частности.

При регистрации *Главного клиента* необходимо задать два параметра:

- Размер очереди (буфера) событий. Данный параметр предназначен для регулирования системой контроллеров при невозможности передачи *Главному клиенту* событий. В случае накопления в буфере более указанного параметром количества событий Драйвер переводит все контроллеры в автономный режим. По-

⁶ Эти условия необходимы для обеспечения надежности и безопасности работы СКУД.

добная ситуация может возникнуть как при медленных линиях связи, так и при некорректном построении программы клиента⁷.

- Признак записи протокола полученных событий (см. п. [Запись протокола «сырых» событий](#)).

Главный клиент после успешного соединения должен выполнить как минимум следующие действия:

- Передать *Серверу* параметры связи с оборудованием (команды создания каналов и контроллеров).
- Проконандовать *Серверу* активизировать созданные каналы и включить опрос на заданных контроллерах.

Только после этого *Клиент*:

- Начнет получать от *Сервера* события о работе системы. Эти события разделяются на события самого оборудования (например, приложен ключ, нажата кнопка) и события об ошибках связи с контроллерами.
- Сможет управлять контроллерами, т.е. включать при необходимости реле, загружать коды ключей и прочее (*Главный клиент*).

В новом сеансе работы с *Сервером* все перечисленные операции должны быть выполнены заново.

6.2. Режимы работы с оборудованием

Для разработчика прикладного ПО важно понимать условия, при которых контроллер работает в комплексе, т.е. **не** принимает решения самостоятельно:

- Опрос контроллера должен производиться с частотой не менее чем один раз в три секунды.
- Команда на получение события должна передаваться с параметром «без перехода в автоном».

Оба этих условия реализованы внутренними механизмами драйвера, от разработчика требуется осмысленно передавать необходимый параметр (*IsAuto*) при выдаче команды «включить опрос» (*PollOn*).

Возможны три основных режима работы с оборудованием:

- Полностью автономный режим. Для его работы необходимо загрузить в память контроллеров коды ключей и права их доступа. После этого прикладная программа может быть остановлена и далее запускаться только при необходимости ввести новые ключи или вычитать события о работе контроллеров (например, для построения отчетов о рабочем времени). Заметим, что для загрузки ключей в память контроллера, последний необходимо перевести в комплексный режим.
- Автономный режим с вычитыванием событий. Режим аналогичен предыдущему (т.е. принятие решений о проходе выполняется самими контроллерами). Однако при этом возможно получение событий в реальном времени. Кроме того, клиентское приложение в любой момент может вмешаться в работу (загрузить ключ, включить реле и прочее). Активизируется командой *PollOn* с параметром *IsAuto=True*.

⁷ Например, при попытке обработки принятых событий в самом событии. Правильно при получении складывать события в буфер, который обрабатывается отдельной процедурой или потоком.

- Комплексный режим, в котором решения принимаются только управляющей программой. Контроллеры при этом только передают события и ждут команды на выполнение. Активизируется командой *PollOn* с параметром *IsAuto=False*.

6.3. Восстановление связи с оборудованием

Если в процессе опроса любой контроллер канала не отвечает на запрос *Сервера* в течение заданного таймаута (параметр *ChannelResponseTimeout*), Сервер будет продолжать опрашивать данный контроллер период времени, указанный параметром *ChannelAliveTimeout*.

Если в течение этого интервала связь с оборудованием не восстановится (т.е. контроллер не ответит), то драйвер прекратит попытки опроса на период параметра *ChannelDeadTimeout*. По прошествии этого времени цикл попыток восстановления связи повторится.

7. Система команд и событий

Драйвер оборудования (*Сервер*) СКУД марки TSS (*Servcont*) представляет собой программу, реализованную под Windows и под Linux.

Клиент осуществляет связь с *Сервером* по TCP/IP протоколу посредством механизма удаленного вызова процедур (RPC), для чего на *Клиенте* указывается имя или IP адрес ПК *Сервера*.

Обмен данными между *Клиентом* и *Сервером* осуществляется путем вызова соответствующих процедур и функций, передающих требуемые данные как параметры вызова.

Практическая реализация работы с *Сервером* подробно расписана в демонстрационном примере *ServcontClient*.

Ниже разъясняются механизмы действия основных функций.

7.1. Запись протокола «сырых» событий

Драйвер имеет возможность писать т.н. протокол «сырых» событий – в том виде, как они приходят от контроллера. Он нужен главным образом для отладки. После отладки системы включать запись в него не рекомендуется.

Протокол ведется в БД формата *SQLite* (файл *coevtlog.db* в папке *..\DATA*), который создается автоматически. Для работы достаточно соответствующей библиотеки (*sqlite3.dll*). Обратите внимание, что путь к указанному файлу не должен содержать русские символы.

Для чтения файла протокола существует команда *srvCoEvtLogSend*. В списке параметров условия фильтрации данных. Чтение может производиться только при неинициализированной системе (не создано ни одного канала). После выдачи запроса данные начинают поступать так же, как события, вычитанные непосредственно из контроллера (т.е. отработывают процедуры *ControllerKey* и прочее).

Для очистки чтения файла протокола применяется команда *srvCoEvtLogClear*.

7.2. Запись ключа в память контроллера

Для записи ключа в память контроллера необходимо задавать код ключа, адрес канала, адрес контроллера, маску портов, номер категории.

Маска портов означает номера портов контроллера (т.е. пунктов прохода), по которым должно быть включено реле (т.е. данный ключ имеет право прохода). Маска должна быть передана как массив из 8 элементов типа `boolean`, в котором `true` указывает на разрешение прохода по соответствующему порту.

Категория – это номер расписания для данного ключа. Для работы контроля времени прохода расписание должно быть загружено в память контроллера. Обратите внимание, что если расписание загружено, но в нем отсутствует категория для конкретного ключа – ключ пропущен не будет.

7.3. Работа с расписанием

Контроллере хранит 16 расписаний. Каждое расписание – это 7 дней недели и т.н. специальный день (например, праздник). Каждый день предусматривает неограниченно число интервалов (например, с 7 до 8, с 12 до 13, с 16 до 17 и т.д.).

Существует список специальных или праздничных дней (до 255). Каждому такому дню ставится в соответствии номер дня недели из расписания, заданного данному ключу. По умолчанию «праздничным» считается восьмой день недели.

Номер расписания – это категория для присвоения ее ключу при записи последнего в память контроллера. Если расписание загружено, но в нем отсутствует категория для конкретного ключа – ключ пропущен не будет.

7.4. Работа с реле

В команде «включить реле» указывается адрес канала, адрес контроллера, номер порта, интервал включения реле. Если в интервале задать 0 реле включиться навсегда. После этого следует применить команду «выключить реле».

8. Приложение 1 Система команд

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
Servcont (префикс srv)	AddChannel	Добавить канал	<p>Host (S) – IP канала (только для IP соединения).</p> <p>IsIp (B) – признак IP соединения.</p> <p>Port (I) – порт для IP соединения.</p> <p>Port (S) – порт для COM соединения (только для соединения через сом-порт).</p> <p>Speed (I) – скорость контроллеров (только для соединения через сом-порт).</p> <p><i>ChannelResponseTimeout</i> (I) – время ожидания ответа от контроллеров (мсек.).</p> <p><i>ChannelAliveTimeout</i>(I) – период попыток восстановления связи с контроллером (сек.).</p> <p><i>ChannelDeadTimeout</i>(I) – время ожидания между попытками восстановления связи (сек.).</p>		Восстановление связи с оборудованием
Servcont	RemoveChannel	Удалить канал	<p>ID (S) – Host : Port (для IP соединения).</p> <p>– Port (для соединения через сом-порт).</p>		
Servcont	ControllChannel	Активировать/деактивировать канал	<p>ID (S) – Host : Port (для IP соединения).</p> <p>– Port (для соединения через сом-порт).</p> <p>Active (B) – True/False</p>	192.168.0.1:5086	

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
Servcont	ChannelList	Получить список каналов	Возврат: Data (S) – перечень каналов.		
Servcont	AddController	Добавить контроллер	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера.		
Servcont	RemoveController	Удалить контроллер	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера.		
Servcont	RemoveController	Удалить контроллер	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера.		
Servcont	PollOn	Включить контроллер в опрос	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. IsAuto (B) – автономный режим. IsReliable (B) – усиленный контроль доставки события (не реализован).	IsAuto – автономный режим с возможностью управления контроллером и вычитывания событий.	
Servcont	Выключить опрос	Выключить опрос	Channel (S) – Host : Port (для IP соединения).	ForceAuto -	

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
	контроллера PollOff	контроллера PollOff	<p>– Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p> <p>ForceAuto (B) – переход в автоном по прекращению опроса (False) или по команде (True, вычитать событие и перейти в автоном).</p>	рекомендуемое значение True.	
Servcont	ChannelList	Получить список каналов	<p>Возврат:</p> <p>Data (S) – перечень каналов.</p>	Data – строка параметров каналов (IP:port или номер сом-порта). Разделитель CLRF (0D0A).	
Servcont	ControllerList	Получить список контроллеров на канале	<p>Channel (S) – Host : Port (для IP соединения).</p> <p>– Port (для соединения через сом-порт).</p> <p>Возврат:</p> <p>Data (S) – перечень контроллеров.</p>	Data – строка адресов контроллеров на канале. Разделитель CLRF (0D0A)	
Servcont	SwitchToAuto	Перевести всю систему в автономный режим			
Servcont	MainClient	Зарегистрироваться как главный	LogControllerEvents (B) – Признак записи протокола событий.		Работа с клиента-

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
		клиент	QueueLimit (I) – размер очереди событий		ми
Servcont	CoEvtLogSend	Получить список сохраненных событий	<p>BEG (S) – дата и время начала выборки.</p> <p>END (S) – дата и время конца выборки.</p> <p>Limit (I) – число доставляемых записей.</p> <p>Offset (I) – смещение (номер первой доставляемой записи).</p> <p>Возврат:</p> <p>Count (I) – число записей в выборке.</p>	<p>Протокол событий сохраняется если включен параметр LogControllerEvents процедуры MainClient.</p> <p>Формат файла протокола – таблица SQLite.</p> <p>Записи выборки доставляются как текущие события контроллеров.</p>	Запись протокола «сырых» событий
Servcont	CoEvtLogClear	Очистить список сохраненных событий			
Controller (префикс	RelayOn	Включить реле	<p>Channel (S) – Host : Port (для IP соединения).</p> <p>– Port (для соединения через сом-порт).</p>		Работа с реле

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
cnt)			Addr (I) – Адрес контроллера. Port (I) – номер порта контроллера. Interval (I) – время включения реле. SuppressDoorEvent (B) – Подавить событие DATA (дверь открыта) True/False.		
Controller	RelayOff	Выключить реле	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Port (I) – номер порта контроллера.		Работа с реле
Controller	KeyExist	Найти ключ	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Key (S) – код ключа. Возврат: Result (B) – True/False. KeyAttr (S) – атрибуты ключа (TControllerKeyAttr).	См. TSSServContAPI.pas	
Controller	EraseKey	Удалить ключ	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт).		

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
			Addr (I) – Адрес контроллера. Key (S) – код ключа.		
Controller	Записать ключ WriteKey	Записать ключ WriteKey	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Key (S) – TControllerKey.	См. TSSServContA PI.pas	Запись ключа в память контрол- лера
Controller	Записать массив ключей WriteAllKeys	Записать массив ключей WriteAllKeys	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Keys (S) – TControllerKey.	См. TSSServContA PI.pas	
Controller	Удалить все ключи EraseAllKeys	Удалить все ключи EraseAllKeys	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера.		
Controller	Прочитать все ключи ReadAllKeys	Прочитать все ключи ReadAllKeys	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Возврат: Count (I) – Число ключей.		

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
			Keys (S) – Список ключей (TControllerKey).		
Controller	ReadAllChips	Прочитать все чипы	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Возврат: Count (I) – Число чипов. Chips (S) – Список чипов (TControllerChip).	Для сигнальных контроллеров 203-О.	
Controller	WriteAllChips	Записать все чипы	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Chips (S) – Список чипов (TControllerChip).	Для сигнальных контроллеров 203-О.	
Controller	ControlChip	Блокировать/разблокировать чип (Снять с охраны / поставить на охрану датчик)	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Chip (I) – Адрес чипа. Active (B) – блокировать (False)/разблокировать (True) чип.	Для сигнальных контроллеров 203-О.	
Controller	EraseAllChips	Удалить все чипы	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт).	Для сигнальных	

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
			Addr (I) – Адрес контроллера.	контроллеров 203-О.	
Controller	Включить таймер Timer	Включить таймер Timer	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Interval (I) – период таймера.	Используется при отладке ПО.	
Controller	Выключить таймер Timer	Выключить таймер Timer	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Interval (I) – период таймера = 0.	Используется при отладке ПО.	
Controller	Генерировать со- бытие таймер GenerateTimerE- vents	Генерировать со- бытие таймер GenerateTimerE- vents	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Count (I) – число событий.	Используется при отладке ПО.	
Controller	ProgId	Номер прошивки ПЗУ контроллера	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт).		

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
			Addr (I) – Адрес контроллера. Возврат: Result (I) – ID прошивки.		
Controller	ProgVer	Версия прошивки ПЗУ контроллера	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Возврат: Result (I) – Версия прошивки.		
Controller	Серийный номер контроллера SerNum	Серийный номер контроллера SerNum	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Возврат: Result (I) – Серийный номер.		
Controller	Получить дату и время контроллера ReadClock	Получить дату и время контроллера ReadClock	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Возврат:		

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
			Result (I) – дата, время (TServcontDateTime).		
Controller	Изменить дату контроллера WriteClockDate	Изменить дату контроллера WriteClockDate	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Value (I) – дата (TServcontDate).		
Controller	Изменить время контроллера WriteClockTime	Изменить время контроллера WriteClockTime	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Value (I) – время (TServcontTime).		
Controller	ReadTimetable	Прочитать расписание	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Возврат: SpecialDays (S) – нестандартные дни (TControllerTimetableSpecialDay). Items (S) – расписание (TControllerTimetableItem).		Работа с расписанием
Controller	WriteTimetable	Записать расписание	Channel (S) – Host : Port (для IP соединения).		Работа с

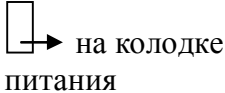
Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
		ние	<p>– Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p> <p>SpecialDays (S) – нестандартные дни (TControllerTimetableSpecialDay).</p> <p>Items (S) – расписание (TControllerTimetableItem).</p>		расписанием
Controller	Удалить расписание EraseTimetable	Удалить расписание EraseTimetable	<p>Channel (S) – Host : Port (для IP соединения).</p> <p>– Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p>		Работа с расписанием
Controller	RestartProg	Перезагрузить ПО контроллера	<p>Channel (S) – Host : Port (для IP соединения).</p> <p>– Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p>		
Controller	EraseAllEvents	Удалить все события	<p>Channel (S) – Host : Port (для IP соединения).</p> <p>– Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p>		
Controller	EventsInfo	Информация о событиях	<p>Channel (S) – Host : Port (для IP соединения).</p> <p>– Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p>		

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
			<p>Возврат:</p> <p>Capacity (I) – емкость стека событий.</p> <p>Count (I) – число событий в стеке.</p>		
Controller	KeysInfo	Информация ключях	<p>o Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p> <p>Возврат:</p> <p>Capacity (I) – емкость стека ключей.</p> <p>Count (I) – число ключей в стеке.</p>		
Controller	PortsInfo	Информация портах	<p>o Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p> <p>Возврат:</p> <p>Result (I) – битовая маска портов.</p>	<p>1100 0000 – 2 порта</p> <p>1111 0000 – 4 порта</p> <p>1111 1100 – 6 портов</p> <p>1111 1111 – 8 портов</p>	
Controller	ReadKeypad	Информация кейпадах	<p>o Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт).</p> <p>Addr (I) – Адрес контроллера.</p>	См. TSSServContA PI.pas	

Категория	Имя	Назначение	Параметры S-string, I-integer, B-boolean	Примечания, примеры	Ссылка на докумен- тацию
			Возврат: Data (S) – tKeypadItems.		
Controller	WriteKeypad	Запись признаков кейпада	Channel (S) – Host : Port (для IP соединения). – Port (для соединения через сом-порт). Addr (I) – Адрес контроллера. Data (S) – tKeypadItems.	См. TSSServContA PI.pas	

9. Приложение 2 Система событий

Категория	Имя процедуры RpcNeedExecute (ProcName)	Назначение	Соединение на контроллере (TSS 209)	Параметры S-string, I-integer, B-boolean	Примечания, примеры
C	ControllerKey	Считан код ключа	1,0,G	TControllerKeyEvent TControllerPortRelayEvent	
C	ControllerButton	Нажата кнопка входа	R,G (нормально разомкнутый)	TControllerKeyEvent TControllerPortRelayEvent	
C	ControllerDoorOpen	Датчик двери (раз- мыкание)	D,G (нормально замкнутый)	TControllerPortEvent TControllerPortRelayEvent	
C	ControllerDoorClose	Датчик двери (за- мыкание)	D,G (нормально замкнутый)	TControllerPortEvent TControllerPortRelayEvent	
C	Controller220V	Отсутствует пита- ние 220 В		Channel (S) – канал. Data (S) – tControllerEvent.	
C	ControllerCase	Открыта крышка контроллера		Channel (S) – канал. Data (S) – tControllerEvent.	
C	ControllerTimer	Событие Timer		Channel (S) – канал. Data (S) – tControllerEvent.	
C	ControllerAutoTimeout	Переход в автоном по таймауту.		Channel (S) – канал.	

Категория	Имя процедуры RpcNeedExecute (ProcName)	Назначение	Соединение на контроллере (TSS 209)	Параметры S-string, I-integer, B-boolean	Примечания, примеры
				Data (S) – tControllerEvent.	
C	ControllerRestart	Перезагрузка внутреннего ПО контроллера		Channel (S) – канал. Data (S) – tControllerEvent.	
C	ControllerStart	Старт контроллера		Channel (S) – канал. Data (S) – tControllerEvent	
C	ControllerStaticSensor	Включение/отключение реле всех портов контроллера при подаче /снятии напряжения	 на колодке питания	Channel (S) – канал. Data (S) – tControllerEvent (0 – реле включены, 16 – реле выключены).	Используется для интеграции с охранопожарными системами.
S	ChannelError	Канальная ошибка		Channel (S) – канал. Time (S) – время события. Class (S) – класс ошибки. Message (S) – сообщение.	
C	ControllerError	Ошибка контроллера		Channel (S) – канал. Time (S) – время события. Class (S) – класс ошибки. Message (S) – сообщение.	
S	ChannelState	Состояние канала		Channel (S) – канал.	

Категория	Имя процедуры RpcNeedExecute (ProcName)	Назначение	Соединение на контроллере (TSS 209)	Параметры S-string, I-integer, B-boolean	Примечания, примеры
				IsReady (B) – состояние.	
S	ChannelPollSpeed	Скорость опроса		Channel (S) – канал. Value (I) – скорость опроса (запросов в сек.)	Передается при изменении скорости более чем на 5%.